PATENT APPLICATION
Do. No. 2705-0726
Client Ref. No. CPOL262371

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Yu Kwong Ng
Serial No.	10/696,467
Confirmation No.	6905
Filed:	October 28, 2003
Examiner:	Craig E. Walter
Group Art Unit:	2188
For:	PARTIAL KEY HASHING MEMORY
Date:	April 8, 2008

REQUESTED
OK to enter
/CEW/ 8 April 2008

Mailstop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT

Summary of Telephonic Interview

In a telephone call with Applicant on April 7, 2008, the Examiner indicated that claims 1-14, 27-33 and 35-38 are considered allowed. Applicant respectfully submits the attached amendment for the Examiner's consideration to facilitate a notice of allowance for all remaining pending and new claims.

Claim amendments begin on page 2.

Remarks begin on page 8

CLAIMS

1. (Previously presented) A method comprising:

storing a plurality of partial keys corresponding to an equal number of original keys in a hash table, wherein storage of the plurality of partial keys requires less memory than storage of the equal number of original keys, and wherein the plurality of partial keys are used to determine hashing conflicts;

applying a hash function to an original key of said equal number of original keys to generate a partial key and a hash value, wherein the hash value includes a number of bits equal to a total number of bits of the original key minus a total number of bits of the partial key;

accessing the hash table according to the hash value;

reading a stored partial key of the plurality of partial keys from the hash table that corresponds to the hash value, wherein the hash value is not stored in the hash table; and

executing a conflict check by comparing the partial key generated from the original key with the stored partial key .

2. (Previously Presented) The method of Claim 1, wherein the partial key from the hash table corresponding to the hash value includes saved bits comprising a consecutive, sequential string of bits that is a subset of the original key where the subset includes a majority of bits of the original key.

3. (Previously presented) The method of Claim 2, wherein the stored partial key comprises a number of bits equal to or more than a number of bits of the original key minus a number of bits of the hash value.

4. (Original) The method of Claim 1, wherein the hash value is implemented by a linear feedback shift register.

5. (Previously presented) The method of Claim 1 further comprising applying a reverse function on the stored partial key and the hash value to generate the original key.

6. (Previously Presented) The method of Claim 1 further comprising the steps of: reading a result from the hash table corresponding to the hash value; and forwarding a packet of data according to the result read from the hash table.

7. (Previously presented) An apparatus comprising:

a hash table which stores a plurality of partial keys used to determine hashing conflicts, wherein the plurality of partial keys correspond to a plurality of original full keys;

a hash function block coupled to the hash table that applies any polynomial to a full key and generates a partial key and a hash value which is used to point to one of the plurality of partial keys stores in the hash table, wherein the plurality of partial keys include saved bits comprising consecutive, sequentially strings of bits derived from the plurality of original full keys, and wherein the hash value includes a number of bits equal to a total number of bits of the full key minus a total number of bits of the partial key; and

a processor that compares one of the plurality of partial keys to the partial key comprising a majority of bits of the full key, wherein the hash value is not saved in the hash table.

8. (Previously Presented) The apparatus of Claim 7, wherein the hash table comprises a hash table size.

9. (Previously Presented) The apparatus of Claim 7, wherein the one of the plurality of partial keys stored in the hash table comprises a number of bits equal to or more than a number of bits of the full key minus a number of bits of the hash value.

10. (Previously presented) The apparatus of Claim 7, wherein the hash function block comprises a linear feedback shift register.

11. (Previously Presented) The apparatus of Claim 10, wherein the linear feedback shift register corresponds to a Galois version.

12. (Previously Presented) The apparatus of Claim 10, wherein the linear feedback shift register corresponds to a Fibonacci version.

13. (Previously Presented) The apparatus of Claim 7 further including a reverse function generator coupled to the hash table, wherein the reverse function generator restores the full key based on the one of the plurality of partial keys stored in the hash table and the hash value.

14. (Previously Presented) The apparatus of Claim 7 further comprising a forwarding engine coupled to the hash table, wherein the forwarding engine forwards a data packet according to information read from the hash table at an address corresponding to the one of the plurality of partial keys stored in the hash table.

15-26. Cancelled

27. (Currently amended) A method comprising:

generating a partial key and a hash value from an original key, wherein ~~where~~ the partial key includes a consecutive subset of a majority of bits of the original key and the hash value includes a number of bits equal to a total number of bits of the original key minus a total number of bits of the partial key;

accessing a hash table including multiple partial keys, wherein ~~where~~ the hash value is not stored in the hash table;

selecting a stored partial key from the hash table that corresponds with the hash value;

comparing the partial key with the stored partial key; and

identifying a hash conflict when the partial key matches the stored partial key.

28. (Currently amended) The method of Claim 27, wherein ~~where~~ the multiple partial keys correspond to an equal number of multiple input keys.

29. (Previously presented) The method of Claim 28, wherein ~~where~~ the multiple partial keys are each selectable according to a different hash value derived from one of the equal number of multiple input keys.

30. (Previously Presented) The method of Claim 27, wherein ~~where~~ the comparing of the partial key comprises reading less data than that contained in the original key.

31. (Previously Presented) The method of Claim 32, wherein ~~where~~ the hash value corresponds to a single entry in the hash table.

32. (Previously presented) The method of Claim 27 including recovering the original key by combining the stored partial key with the hash value.

33. (Previously presented) The method of Claim 32, wherein ~~where~~ the original key is recovered by a reverse linear feedback shift register.

34. (Currently amended) A system comprising:

a hash function ~~configured to generate~~ that generates a hash value and a partial key from an input key, wherein ~~where~~ the partial key includes a consecutive sequential string of bits derived from the input key and the hash value includes a number of bits equal to a total number of bits of the input key minus a total number of bits of the partial key;

a hash table including stored partial keys that correspond to an equal number of input keys, wherein ~~where~~ the partial key includes less data bits than the input key and wherein ~~where~~ the hash value is not stored in the hash table; and

a processor ~~configured to~~ that:

~~identify~~ identifies the partial key that is associated with the hash value;

~~compare~~ compares the partial key to the stored partial keys; and

~~identify~~ identifies a hash conflict when the partial key matches one of the stored partial keys.

35. (Previously Presented) The system of Claim 34, wherein ~~where~~ the stored partial keys include a majority of data bits of the corresponding input keys.

36. (Previously presented) The system of Claim 34, wherein ~~where~~ the partial key includes multiple consecutive sequential strings of bits derived from the input key, and the multiple consecutive sequential strings of bits are separated by the bits of the hash value.

37. (Previously Presented) The system of Claim 34, wherein ~~where~~ the hash value includes an address location associated with the input key.

38. (Previously Presented) The system of Claim 34, wherein ~~where~~ the hash value includes bits from the input key that are not included in any of the stored partial keys.

39. (New) A computer-readable medium having instructions stored thereon, wherein when the instructions are executed by at least one device, they are operable to:

generate a partial key and a hash value from an original key, wherein the partial key includes a consecutive subset of a majority of bits of the original key and the hash value includes a number of bits equal to a total number of bits of the original key minus a total number of bits of the partial key;

select a stored partial key from a hash table that corresponds with the hash value, wherein the hash value is not stored in the hash table;

compare the partial key with the stored partial key; and

identify a hash conflict when the partial key matches the stored partial key.

40. (New) The computer-readable medium of Claim 39, wherein the instructions are further operable to:

read a result from the hash table corresponding to the hash value; and

forward a packet of data according to the result read from the hash table.

41. (New) The computer-readable medium of Claim 39, wherein the stored partial key comprises a number of bits equal to or more than a number of bits of the original key minus a number of bits of the hash value.

42. (New) The computer-readable medium of Claim 39, wherein the hash table comprises a plurality of partial keys that correspond to an equal number of input keys, and wherein the plurality of partial keys are each selectable according to a different hash value derived from one of the input keys.

43. (New) The computer-readable medium of Claim 39, wherein the instructions are further operable to:

recover the original key by combining the stored partial key with the hash value.

44. (New) The computer-readable medium of Claim 39, wherein the hash value corresponds to a single entry in the hash table.

REMARKS

Claims 1-14 and 27-38 are pending.

Claims 1-14 and 27-33 and 35-38 are allowed.

Claim 27-31 and 33-38 are amended, herein.

Claims 39-44 are new. No new matter is added.

Claim Amendments

Claims 27-31 and 33-38 are amended for grammatical purposes, and to provide consistency with claim language recited by remaining pending claims. The claim amendments are not being made to overcome any cited reference, and Applicant neither intends nor believes that these amendments operate to narrow the scope of the claims.

New claims

Support for the new claims 39-44 may be found variously at page 10 lines 11-25, page 11 line 11 to page 12 line 15, etc. One skilled in the art would appreciate that the instructions recited by claims 39-44 may be stored on computer-readable medium according to page 20 line 15-17 describing the applicability of the hash functions used in software programming, and further that example embodiments of such instructions may be readily determined from the equations and logic provided throughout the specification (e.g. page 13 to page 19). See also page 2 line 16-24 identifying device 201 and wherein "information that is contained in memory 203 specifies which of the output ports 01-04 that particular packet is to be routed." Several types of memory, including CAM, DRAM and RAM are further described on page 3-4.

Conclusion

For the foregoing reasons, reconsideration and allowance of claims 1-14 and 27-44 of the application as amended is requested. The Examiner is encouraged to telephone the undersigned at (503) 224-2170 if it appears that an interview would be helpful in advancing the case.

Customer No. 73552

Respectfully submitted,

STOLOWITZ FORD COWGER LLP

Bryan Kirkpatrick

Bryan D. Kirkpatrick
Reg. No. 53,135

STOLOWITZ FORD COWGER LLP
621 SW Morrison Street, Suite 600
Portland, OR 97205
(503) 224-2170